

Solving a Real World Problem

Improving conditions to foster innovation targeting human beings.

Bachelor's Thesis

submitted in conformity with the requirements for the degree of

Bachelor of Science in Engineering (BSc)

Bachelor's degree programme **Software Design**

FH JOANNEUM (University of Applied Sciences), Kapfenberg

Supervisor: T. Truth

Submitted by: Julia Feelgood

November 2023

[TODO: Specify the title, sub title, place, date, study, language, your name, and advisor in the main *thesis.tex* file.

Finally, remove all *TODOs* within your \LaTeX source code.]

Abstract

Your text here... [TODO: Write the abstract in English and in German, called *Zusammenfassung*. Describe in about 250 to 350 words the problem, the innovation, the method, the results and implications.]

Zusammenfassung

Ihr Text beginnt hier... [TODO: Die Zusammenfassung sollte das gesamte Werk enthalten, also das spannende Problem, den gewählten – neuartigen – Lösungsansatz und natürlich vor allem die erreichten Resultate.]

Contents

List of Figures	ii
List of Tables	iii
1 Writing a Thesis Using L^AT_EX	1
1.1 Hints on Scientific Writing	1
1.2 L ^A T _E X Formatting Hints	4
1.3 Sentences and Praragraphs	6
1.4 Red Thread	6
2 Introduction	8
2.1 Problem Statement	8
2.2 Research Questions	8
2.3 Hypothesis	8
2.4 Method	9
3 Related Work	10
4 Concept	11
5 Implementation	12
6 Evaluation	13
7 Conclusion and Outlook	14
Bibliography	17

List of Figures

1.1 Logo at the Train Engine 4

List of Tables

1.1	Fancy Table	5
1.2	Simple Grey Table	5

Writing a Thesis Using L^AT_EX

“Research is formalised curiosity. It is poking and prying with a purpose.”

[Zora Neale Hurston]

Each chapter should start with a short explanation what is inside the upcoming chapter and why it has been included (at this position) in your work: This template shall provide some considerations, text examples and formatting hints for your Bachelor’s or Master’s thesis in L^AT_EX.

1.1 Hints on Scientific Writing

Some recommendations on writing an abstract and about including references to related work.

1.1.1 How to Write an Abstract

Learn from others and read many papers¹ related to your work. Finally, you might ensure that your abstract contains:

- English & German Version (250 – 350 words each)
- Background / motivation / problem statement
- Methods / procedure / approach
- Results / findings / product
- Conclusion and implications

¹For examples, read the abstract of paper <https://arxiv.org/abs/1609.03677>.

It is easier to write the Abstract when the rest of your paper is finished.

1.1.2 Research Resources

For literature research use e.g. *ACM Digital Library* (ACM, Inc. (Association for Computing Machinery), 2017) or *IEEE Xplore Digital Library* (IEEE (Institute of Electrical and Electronics Engineers), 2021). You might start your search within the scientific databases <http://dl.acm.org/> or <http://ieeexplore.ieee.org/>. Full-text PDF download is available from within the FH JOANNEUM network.

1.1.3 Citation Styles

Harvard citation style is implemented in this template. For information about a topic like RFID paraphrased in your own words (cf. Batina et al., 2012, p. 317) do not forget to use *cf.* and – if available the relevant page number(s) – along with parenthetical cite `\parencite`. Direct quotations would not need the *cf.* The abbreviation *cf.* is short for Latin *confer* meaning compare. The abbreviation *p.* is short for *page*, and *pp.* is short for *pages*. If you need to use the title of a reference, for example the RFID Authentication Protocol by “A Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation” you might use `\citetitle`. For references without parentheses such as – find more in Li, Baudisch & Hinckley, 2008 – just use `\cite` or, if year should be in parentheses, Batina et al. (2012) `\textcite`.

Note the use of *ibid.* *Ibid* is short for Latin *ibidem* meaning in the same place.

In German *ebd* or *ebenda* is used. *Ibid* is used for referencing (several pages of) the same resource subsequently. For example, see (cf. *ibid.*, p. 317) and (cf. *ibid.*, pp. 321–323) (cf. *ibid.*, p. 399).

You might cite URLs, e.g. about (tools for checking) Accessibility (cf. Google Inc., 2016, 2017), as online resources with a date of your last visit.

1.1.4 Bibliography Entries

Compile your resources you want to cite in a file with so called bibliography entries *bib entries*. Readers must be able to trace back and verify each and every source. Make sure, readers can find the given resources quick and easily.

The required information to provide might differ between the kind of resources. Every entry needs information on author(s), title, and year. For books you need to add the

publisher information and the ISBN. For research papers (from scientific databases such as IEEE or ACM) one need to add the conference title, location and the Digital Object Identifier (DOI). For the DOI and International Standard Book Number (ISBN) numbers, links are automatically generated by L^AT_EX, hence no full link (Universal Resource Locator (URL)) must be specified.

1.1.5 Scientific Writing Resources

Selected resources about scientific working:

- Writing for Computer Science: ...elements of good writing – clarity, simplicity, accuracy, and organization ...by Zobel, 2004.
- “*What makes good research in software engineering?*” : ...e.g. find out ways of validating your findings, your results ...by Shaw, 2002.
- Case Study Research: ...offers comprehensive coverage of the design and use of the case study method as a valid research tool ...by Yin, 2013.
- The Elements of Style: ...first edition about 1935; includes a list of valuable recommendations: be clear, do not overwrite ...by Strunk & White, 2000.
- How to Design and Report Experiments: ...Planning an Experiment, Experimental Designs, Descriptive Statistics, Inferential Statistics ...Answering the Question ‘So What?’ ...by Field & Hole, 2003.
- The Craft of Research: ... What Is Research? Creating a Relationship with Your Reader: Your Role, Finding a Good Research Problem ...by Booth, Colomb & Williams, 2008.
- The Craft of Scientific Writing: ...your writing is the principle way in which people learn about your work. When you communicate well, you receive credit for your ...by Alley, 1998.
- Wie man eine wissenschaftliche Abschlußarbeit schreibt: Doktor-, Diplom- und Magisterarbeit in den Geistes- und Sozialwissenschaften: ... Warum muss man eine wissenschaftliche Abschlussarbeit schreiben und was ist sie? ...by Eco, 2010.
- Writer’s Handbook: ...you will find many instructional materials we’ve developed for our Writing Center teaching: Planning and Writing Research Papers, Creating an Argument, ...by The Writing Center, 2004.

Better take a look at those references *before* starting to write your thesis.



Figure 1.1: Note the logo attached to the train engine spotted in Kapfenberg main station.

1.2 L^AT_EX Formatting Hints

Selected L^AT_EX examples are included to give an impression of how you could add tables, diagrams, or figures to your text. Use *bullet lists* and *emphasised text* and *named paragraphs* sparsely.

Background. In the section on the background describe the prerequisites for your work.

Terms and definitions. Technical terms should be explained if necessary. Abbreviations are summarised at the end of the thesis in Chapter 7 “Acronyms”. The abbreviations are defined in advance using `\acro{ }{ }`. Within the text `\ac{ }` is used. For example, `\ac{ABI}` and `\ac{MITM}` occur in text as Application Binary Interface (ABI) and Man-In-The-Middle (MITM). If `\ac{ABI}` is used again, only the acronym ABI is printed (as hyperlink though).

Visual elements. To support your readers, include visuals such as diagrams and other graphics (see Figure 1.1). Note the short title used for the list of figures.

Version	Description	Author(s)	Date
1.0	Initial	Ohrt	July 15, 2014
1.1	Filled section “Open Issues”	Ohrt	July 16, 2014
1.2	Added section “Restrictions”	Ohrt	September 15, 2014
1.3	Dynamic fields for “BA” and “MA”	J.F.	September 15, 2018
1.4	Restructuring many sections	J.F.	November 127, 2019

Table 1.1: Olive green heading used for this fancy table.

Error	Solution
Java.lang.OutOfMemoryError: PermGen space (32-/64-bit issue)	-XX:MaxPermSize=1024M
Error occurred during initialization of VM <i>or</i> Could not reserve enough space for object heap	increase or remove -Xms value e.g. -Xms128m -Xmx512m (Eclipse default: -Xms40m -Xmx512m)

Table 1.2: A more or less simple grey table. Better try to put tables and figures at *top*[t] or at the *bottom*[b] of a page. With *page*[p] you can put them on a separate page. Avoid location specifier *here*[h].

Code listings require the `listings` package which, in turn, requires some settings. This is, because the defaults do not fit all purposes; see command `\lstset{}` in preamble of this template. Additionally, the package *courier* should be used because the defaults do not provide for proper syntax highlighting². Very small code snippets `func main(){...}` can be marked with `\lstinline{}` in the text.

In order to see what’s possible when formatting tables – here are two fancy tables, see Table 1.1 and Table 1.2 which show demo data. Preferable, use `tabularx`, because the parameter *X* allows to define columns of dynamic size. Online table generators³ can create the L^AT_EXsource for you.

Always reference listings in text, such as Listing 1.2. Use line numbers to help the reader to find relevant parts within given code. Referenced listings, tables and figures are written in uppercase first letter: *Listing X*, *Table Y* and *Figure Z*.

²Find the full list of supported programming languages at https://www.overleaf.com/learn/latex/Code_listing.

³Online table generator <https://www.tablesgenerator.com>.

1.2.1 Prototype

Find in Listing 1.2 an example of the JavaScript closure. Only selected (relevant) parts of the original source code have been included. That allows to extract and display parts of working code!

Hints on listings: use environment `samepage` for not breaking the listing in multiple parts and not spreading the listing over multiple pages (see Listings 1.1 and 1.2). Do not forget to make the listings float with `float=tp`.

```

1 void main(int argc, char *argv[])
2 {
3     printf("Hello world!");
4 }

```

Listing 1.1: No programming language for syntax highlighting is specified, hence the default we specified in `lst`, i.e. `C`, is taken.

Mathematical expressions are rendered beautifully by L^AT_EX. Now enjoy the first Maxwell equation $\text{rot}\vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t}$. Simple mathematical expressions can be written inline enclosed between `$`, such as $ax^2 + bx + c = 0$. Important equations should stand out and be put into dedicated `equation` environment. They get numbered automatically.

$$\frac{a^2}{b-c} = 8 \tag{1.1}$$

1.3 Sentences and Praragraphs

For better readability, try to structure long text into paragraphs. Insert a newline between paragraphs. paragraphs should not be too short, they are expected to contain several sentences.

1.4 Red Thread

At the end of each chapter you might sum up the contents of the chapter in a sentence or two. Then you might tell the reader what will be presented in the upcoming section (to make her/him curious).

[TODO: You have seen how L^AT_EX works; now remove this Chapter.]

```
1 print("we go in for closures now :)")
2
3 function nextNumber(){
4     var startWith=222;
5     function nxt(){
6         startWith+=1 // !!
7         return startWith
8     }
9     return nxt
10 }
11 var nn = nextNumber()
12 print("In latex long source code lines might force a line wrap if the
    ↪ parameter breaklines in lstset is set.")
13 print( nn() )
14 print( nn() )
```

Listing 1.2: Demo implementation of a JavaScript *Closure*.

Chapter 2

Introduction

Your text here... [TODO: Describe the kind of problem at hand? The problem is relevant in which context? What does not work well at the moment? What do people need? Describe the background, the prerequisites for your work. Optionally, add terms and definitions whenever they might not be clear to a fellow student.]

2.1 Problem Statement

Your text here... [TODO: What is the overall problem? Give examples. Motivate! Compared to existing solutions for the problem at hand, why does someone need a better, faster, and somewhat different one?]

2.2 Research Questions

Your text here... [TODO: Focus on one or two main research questions and detail on them.]

2.3 Hypothesis

Your text here... [TODO: State a hypothesis – a rough idea – of how you think a solution might look like. Explain, how to possibly solve a given problem.]

2.4 Method

Your text here. . . [TODO: Your structured, academic approach¹ to find a solution. When you needed (large) data sets for you work, explain how you collected and filtered raw data. For the validation (see Section Evaluation 6) you want to describe the criteria for objective measurement already here.]

¹Find an extensive explanation of how to write a *Method* section at <http://www.mrcophth.com/publishorperish/methods.html>.

Chapter 3

Related Work

Your text here... [TODO: Describe the work of other research teams and noteworthy approaches related to your work. State what is different to your solution.]

[TODO: Related literature might/should contain:
theoretical foundations,
definitions of key terms,
technologies, techniques,
and/or a literature review]

[TODO: Note on the size and quality of your bibliography:
BA about 30-40 references
MA about 60-100 references]

[TODO: Furthermore, check:
Are the reference (too) old?
Did you include papers from scientific databases, such as ACM or IEEE?
Can the reader find your sources? (e.g. check if you named the publisher for books, or specified DOIs for scientific papers)]

Chapter 4

Concept

Your text here... [TODO: Describe an overall concept of a solution, which could possibly solve a given problem. Design a novel solution and visualise the architecture and relevant (data) flows. Compare and relate your approach to possible alternatives and argue why and in which way(s) the suggested solution(s) will be better.]

Chapter 5

Implementation

Your text here... [TODO: Describe what is relevant and special about your working prototype. State how single features help to solve problem(s) at hand. You might implement only the most relevant features. Features you select from your prioritised feature list assembled in Chapter 4. Focus novel, difficult, or innovative aspects of your prototype. Add visuals such as architectures, diagrams, flows, tables, screenshots to illustrate your work. Select interesting code snippets, e.g. of somewhat complicated algorithms, to present them as source code listings.]

Chapter 6

Evaluation

Your text here... [TODO: Describe (proof) how your implementation really solved the stated problem. I.e. accept or reject your hypotheses. Provide a range of input data sets. Run experiments and gather the output (of tools) to meter your prototype. For the analysis, collect the measurement-data, process (e.g. filter) data and interpret the data. Include an interpretation of the work. What do the results mean to you? State current limitations of your solution. Give (personal) interpretation where suitable. Your own opinion is relevant, but must be marked clearly as such.]

Conclusion and Outlook

Your text here... [TODO: Sum up the results achieved. Suggest further research by explaining how others could built on your results.]

Acronyms

[TODO: Add acronyms (abbreviations) and their long version. In the text the first occurrence will show the full description, further occurrences will show the acronym only.]

ABI	Application Binary Interface
DOI	Digital Object Identifier
ISBN	International Standard Book Number
MITM	Man-In-The-Middle
URL	Universal Resource Locator

[TODO: Finally, check the bibliography, because readers must be able to trace back and verify each and every source. Are you sure, that everyone can find the given resources with the information you supplied? Besides author(s), title and year, for books you need the publisher information and the ISBN, for IEEE/ACM research papers add the conference/journal title, location and the DOI.]

Bibliography

- ACM, Inc. (Association for Computing Machinery) (2017). *ACM Digital Library*. Available from: <https://dl.acm.org> [May 17, 2017] (cit. on p. 2).
- Alley, M. (1998). *The Craft of Scientific Writing*. 3rd ed. Springer. ISBN 0387947663. Available from: <http://www.writing.eng.vt.edu/csw.html> (cit. on p. 3).
- Batina, L., S. Seys, D. Singelée & I. Verbauwhede (2012). “Hierarchical ECC-Based RFID Authentication Protocol”. In: *Proceedings of the 7th International Conference on RFID Security and Privacy (RFIDSec’11)*. Springer-Verlag, pp. 183–201. ISBN 9783642252853. DOI: 10.1007/978-3-642-25286-0_12 (cit. on p. 2).
- Booth, W. C., G. G. Colomb & J. M. Williams (2008). *The Craft of Research*. 3rd ed. University of Chicago Press (cit. on p. 3).
- Eco, U. (2010). *Wie man eine wissenschaftliche Abschlußarbeit schreibt: Doktor-, Diplom- und Magisterarbeit in den Geistes- und Sozialwissenschaften*. 13th ed. facultas wuv (UTB). ISBN 9783825215125. Available from: <http://books.google.com/books?id=Y5c7cgAACAAJ> (cit. on p. 3).
- Fernández-Mir, A., R. Trujillo-Rasua, J. Castellà-Roca & J. Domingo-Ferrer (2012). “A Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation”. In: *Proceedings of the 7th International Conference on RFID Security and Privacy (RFIDSec’11)*. Springer-Verlag, pp. 147–162. ISBN 9783642252853. DOI: 10.1007/978-3-642-25286-0_10 (cit. on p. 2).
- Field, A. P. & G. Hole (Feb. 2003). *How to Design and Report Experiments*. Sage Publications. ISBN 0761973826 (cit. on p. 3).
- Google Inc. (Oct. 6, 2016). *Accessibility Scanner*. Available from: <https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor> [Feb. 8, 2017] (cit. on p. 2).

- Google Inc. (Feb. 8, 2017). *Testing Your App's Accessibility*. Available from: <https://developer.android.com/training/accessibility/testing.html> [Feb. 8, 2017] (cit. on p. 2).
- IEEE (Institute of Electrical and Electronics Engineers) (2021). *IEEE Xplore Digital Library*. Available from: <https://ieeexplore.ieee.org/> [May 17, 2021] (cit. on p. 2).
- Li, K. A., P. Baudisch & K. Hinckley (Apr. 2008). "Blindsight: Eyes-Free Access to Mobile Phones". In: *Proc. 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI2008) (CHI 2008)*. ACM, pp. 1389–1398. ISBN [9781605580111](#). DOI: [10.1145/1357054.1357273](#) (cit. on p. 2).
- Shaw, M. (2002). "What makes good research in software engineering?" In: *International Journal on Software Tools for Technology Transfer* 4.1, pp. 1–7 (cit. on p. 3).
- Strunk Jr, W. & E. B. White (2000). *The Elements of Style*. 4th ed. Pearson Higher Education. ISBN [020530902X](#) (cit. on p. 3).
- The Writing Center (2004). *Writer's Handbook*. University of Wisconsin, Madison. Available from: <http://www.wisc.edu/writing/Handbook/index.html> (cit. on p. 3).
- Yin, R. K. (2013). *Case Study Research*. Sage Publications. ISBN [9781452242569](#) (cit. on p. 3).
- Zobel, J. (Apr. 2004). *Writing for Computer Science*. 2nd ed. Springer. ISBN [1852338024](#) (cit. on p. 3).